

<https://angular.dev/guide/forms/reactive-forms>

Reactive Forms!

```
@Component({
  selector: 'app-contact',
  imports: [ReactiveFormsModule],
  templateUrl: './contact.component.html',
  styleUrls: ['./contact.component.css'],
})
export class ContactComponent {
```

```
<form [formGroup]="messageForm"
  (submit)="sendMessage()">
  <fieldset>
    <legend>Contact</legend>
    <ul>
      <li>
        <label for="name">Name</label>
        <input
          name="name"
          type="text"
          id="name"
          placeholder="Your name"
          formControlName="name"
        />
      </li>
```

Reactive Forms

- ***Observables*** naudojami stebėti reikšmių pokyčius ir atnaujinimus.
- ***Immutable*** – kiekvieno pokyčio metu grąžinama nauja formos būseną.
- Leidžia atlikti **validaciją kode** – lengviau pritaikyti individualius sprendimus.

Reactive Forms

- Sudaryta iš **FormControl** (vienas form control), **FormGroup** (formos valdiklių map) ir **FormArray** (**FormGroup** masyvai).

```
messageForm = new FormGroup({  
  name: new FormControl(''),  
  email: new FormControl(''),  
  text: new FormControl(''),  
});
```

- Apibrėžiama kode – tik atvaizduojama šablone.

- **[formGroup]** priskiriamas *form* elementui:

```
<form [formGroup]="messageForm" (submit)="sendMessage()">
```

- **formControlName** atributas susieja HTML formos valdiklį su kodu:

```
<input  
  name="name" type="text"  
  id="name"  
  placeholder="Your name"  
  formControlName="name"  
>
```

- **FormBuilder** gali būti *inject'inamas*, kad būtų paprasčiau kurti formas:

```
messageForm =  
this.#fb.group({  
  name: '',  
  email: '',  
  text: '',  
});
```

Reactive Form Validacija

```
name: ['', [Validators.required, Validators.pattern('Jurgis')]],
```

```
get messageName() {  
  return this.messageForm.get('name');  
}
```

```
<li>  
  <label for="name">Name</label>  
  <input  
    name="name"  
    type="text"  
    id="name"  
    placeholder="Your name"  
    formControlName="name"  
  />  
</li>  
<div  
  [hidden]="messageName?.valid ||  
messageForm.controls['name'].pristine"  
>  
  @if (messageName?.errors?.['required']) {  
    <span>Expense name is required</span>  
  } @else if (messageForm.controls['name'].errors?.['pattern']) {  
    <span>Tik Jurgis</span>  
  }  
</div>
```

Template driven to Reactive

```
<li>
  <label for="name">Name</label>
  <input
    name="name"
    type="text"
    id="name"
    [(ngModel)]="model.name"
    required
    placeholder="Your name"
    #messageName="ngModel"
    pattern="Jurgis"
  />
</li>
<div [hidden]="messageName.valid || messageName.pristine">
  @if(messageName.errors?.['required']) {
    <span>Message name is required</span>
  } @else if (messageName.errors?.['pattern']) {
    <span>Tik Jurgis</span>
  }
</div>
```

Template driven to Reactive

```
<li>
  <label for="name">Name</label>
  <input
    name="name"
    type="text"
    id="name"
    placeholder="Your name"
    formControlName="name"
  />
</li>
<div
  [hidden]="messageName?.valid || messageForm.controls['name'].pristine"
>
  @if (messageName?.errors?.['required']) {
    <span>Expense name is required</span>
  } @else if (messageForm.controls['name'].errors?.['pattern']) {
    <span>Tik Jurgis</span>
  }
</div>
```

Task #2 - Reactive Forms intro

- Konvertuokite **TasksComponent** *template-driven* formą į *reactive*.
- Template validatoriai >> reactive validatoriai.